ABSTRACT

In order to manipulate video objects in image sequences, we need to segment the image into a set of objects and the background. For video segmentation, various algorithms have been proposed. In a typical spatio-temporal segmentation scheme, the change detection mask is obtained by a statistical decision on the difference image between two successive frames. Therefore, this scheme could not detect slowly moving video objects since slowly moving object results in only small difference values. In order to overcome this problem, we could exploit history information with some memory. In this paper, we propose a new segmentation scheme using the history of motion information. Although performances of conventional spatio-temporal segmentation schemes depend on the speed of the moving objects, our proposed scheme reveals its robustness to the speed of the moving objects.

1. INTRODUCTION

Video processing techniques have been developed, a new functionality of content-based interactivity is needed for acceptable manipulation of video objects. However, traditional video standards such as MPEG-1, MPEG-2, H.261, H.263 are low level techniques in the sense that no segmentation or analysis is required.

In recent years, we have seen that MPEG-4 has become a very promising standard for multimedia applications. MPEG-4 enables content-based functionalities by introducing the concept of video object planes (VOP's). In order to access an image based on its contents, we should be able to segment the image into a set of objects or regions and to estimate their parameters such as color, shape, and motion.

Knowledge about the shape of video objects in a scene enables us a better image reconstruction, especially at object boundaries. It improves coding efficiency and makes it possible to transfer multimedia data at a very low bit rate.

A frame of the input sequence is segmented into arbitrary region images such that each video object describes one semantically meaningful object or video content of interest. A video object layer is assigned to each video object containing shape, motion, and texture information.

An intrinsic problem of video object segmentation is that objects of interest are not homogeneous with respect to low-level features, such as color, intensity, or optical flow.

Among the conventional approaches for segmentation, the morphological segmentation technique is particularly interesting to us because it employs morphological tools which are very attractive for dealing with object-oriented features, such as size and contrast. This scheme produces a gradient image using morphological operations, and labels each region using a watershed algorithm [1]. After testing each region with a certain criterion, this scheme decides which region belongs to a video object. However, morphological filters operating only on luminance values may cause false contours. Particularly, segmentation is largely affected by the shade information.

Another interesting approach is change detection based on a statistical model for the video. In this method, we examine luminance differences between two successive frames and recognize moving objects by significant luminance change. However, an inherent difficulty when evaluating difference images is imposed by the presence of noise, which give rise to intensity changes covering moving areas as well as stationary ones. This scheme, by test statistics computed from local gray level differences, thus suffers from the dilemma of either causing false alarms or failing to detect considerable parts of genuinely moving objects [2]. In addition, this scheme is dependent on the false alarm rate set by user.

We note that the morphological segmentation scheme exploits only the spatial information and the statistical model-based change detection approach utilizes only the temporal information. Recently several spatio-temporal algorithms, which are in principle combinations of the above two approaches, have been proposed to attack the video segmentation problem. Currently, MPEG-4 FCD selected a typical spatio-temporal algorithm which is contributed by ETRI, FUB, and UH [4]. These methods extract edge information from the morphological operation, and obtain information for moving regions from the change detection mask [3, 4, 5]. In a typical spatio-temporal segmentation scheme, change
detection mask is obtained by a statistical decision. This scheme declares video objects against the background based on the frame differences. Therefore, this scheme could not detect slowly moving video objects since slowly moving object results in only small difference values. The performance of this scheme may be sensitive to the speed of video objects.

In order to overcome this problem, we could exploit the history information. In this paper, we propose a new segmentation scheme using the history information based on region tracking. Our proposed scheme reveals its robustness to the speed of the moving objects due to the proper exploitation of the history information.

2. SPATIO-TEMPORAL SEGMENTATION

In this section, we review a typical spatio-temporal segmentation algorithm. Fig. 1 describes a typical spatio-temporal algorithm which is selected by MPEG-4 FCD [3]. In Fig. 1, the spatio-temporal algorithm consists of the spatial part, the temporal part, and the foreground/background decision part which uses temporal information and spatial information. The spatial segmentation part consists of simplification based on morphological filters, morphological gradient approximation, and region marking by watershed algorithm. The temporal part consists of global motion estimation/compensation, and change detection mask.

In the simplification step, images are simplified to make the image segmentation easier. Morphological open-close and close-open by reconstruction filters are used for image simplification. These filters remove regions that are smaller than a given size, but preserve the contours of the objects in the image [6]. The size of the regions eliminated is dependent on applications.

The spatial gradient of the simplified image is approximated by using a morphological gradient operator [7]. The spatial gradient can be used as an input to the watershed algorithm in order to partition an image into homogeneous intensity regions. The resulting gradient image actually exhibits many noisy gradients, which can cause the watershed algorithm to oversegment the image. In order to alleviate this problem, the gradient images are thresholded by a given value. Small gradient values which are less than the threshold value are set to zero.

To obtain the contours of objects, the watershed algorithm should work on the gradient of the image signal. The watershed algorithm derives from topographic works where the catchment basins and their dividing lines, which are called watershed lines, have been extensively studied [1]. As can be seen in Fig. 2, watershed lines partition the space by associating each catchment basin to its local minima. A watershed algorithm can be performed by immersion simulation in digital spaces [1]. First of all, we pierce on the holes in each regional minimum. We then slowly immerse our surface into a lake. Starting from the minima of lowest altitude, the water will progressively fill up the different catchment basins. At each pixel where the water coming from two different minima would merge, we build a dam. At the end of this immersion procedure, each minimum is completely surrounded by dams, which delimit its associated catchment basin. The final result of the algorithm is a tessellation of the input image into its different catchment basins.

For the global motion estimation, we use the affine parameter motion model. The affine motion model of 6 parameters is expressed as follows:

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} = \begin{bmatrix}
    a_{1} & a_{2} \\
    a_{4} & a_{5}
\end{bmatrix} \begin{bmatrix}
    x \\
    y
\end{bmatrix} + \begin{bmatrix}
    a_{3} \\
    a_{6}
\end{bmatrix}
\]

(1)

To obtain 6 parameters, we should find the local motion vec-
If the amount of the global motion is large, a robust motion compensation is performed by Eq. (1).

The change detection mask step, we make difference images between successive frames. The probability density function of the difference images is assumed with the Gaussian distribution or Laplacian distribution. To make the detection algorithm more reliable, the decision should not be based on the gray level difference at a single pixel location only, but on local samples comprising several differences. Thus our sufficient statistic is the square sum of gray level differences inside the local window. In this case, both Laplacian distribution and Gaussian distribution are converted to \( \chi^2 \)-pdf, and the difference is the degree of freedom \( p \). To make a change detection mask, we can employ a Neyman-Pearson detector, ML detector, or MAP detector.

Finally, in the foreground/background decision step, we should check how many portions of a labeled region from watersheds are occupied with the regions declared as changed through the statistical test. If the result is larger than a threshold, this labeled region is declared as foreground. Otherwise, it is regarded as background.

The results of this scheme are shown in Fig. 3 and Fig. 4. This method could not detect a slowly moving video object, since the slowly moving object result in small difference values. For good results, the speed of video object should be larger than some threshold value.

3. REGION TRACKING WITH HISTORY INFORMATION

In the previous section, we mentioned the problem of the slowly moving video object. In this section, we consider a simple approach and a region tracking method in order to avoid it. These techniques use history information.

The temporal segmentation techniques use the statistical estimation and detection methods. Since those techniques have inherently false alarm probability and missing probability, if the speed of video object is slow, some regions of video object might be declared as background due to the missing probability. To avoid this problem, we employ a simple method that exploits history information with some memory. In this method, if a labeled region in the current frame includes some changed region and this labeled region is determined as the background based on the CDM (change detection mask), we check how many portions of this region are occupied with the video object at the same locations in the previous frame. If the result is larger than a threshold value, this region is declared as foreground. This method works well on image sequences of the head and shoulder type, such as AKIYO and MOTHER_DAUGHTER. However, the performance of this scheme is dependent on the information, which can provide robust performance for various input sequences. Our scheme consists of three steps.

In the first step, we decide which region belongs to the video object using a statistical test on the difference image between two successive frames and the boundary information of each labeled region from watersheds in the current frame as described in the previous section. Our experiment employs the Neyman-Pearson detector as a statistical detector with size \( \alpha = 0.05 \). After matching the changed region to a labeled region, the labeled region matched in current frame is declared as foreground with the boundary of the labeled region.

Second, we project each region declared as the foreground region in the previous frame to the current frame through a motion estimation technique. In this step, we employ the polygon-matching algorithm for each foreground labeled region in the previous frame because each region has an arbitrary shape. We select the sum of absolute differences of pixel values inside the polygon as the matching criteria, which is normalized by the number of pixels inside the polygon.

The third step is boundary fitting. The shape of the region projected to the current frame is different from that of the labeled region at the location founded by the motion vector in the current frame. In this case, if the projected region covers the most part of the region labeled by watershed in the current frame, the labeled region is also declared as the foreground. In this method, if a region has a motion vector with a large normalized sum of absolute differences, this region should be discarded. The results of the proposed scheme are shown in Fig. 3(d) and Fig. 4(d). The performance of the proposed scheme has the robustness for various input image sequences.

4. SIMULATION RESULTS

Our scheme is applied to several QCIF (176x144) images. Fig. 3 shows the results for the MOTHER_DAUGHTER sequence and Fig. 4 shows the results for the HALL_MONITOR sequence. The feature of Fig. 3 is that the bodies of objects have no motion. The feature of Fig 4 is that a video object has relatively fast motion. Fig. 3(b) and Fig. 4(b) are results with no history information. Fig. 3(c) and Fig. 4(c) are results of the simple method which use the information at the same location in the previous frame. Fig. 3(d) and Fig. 4(d) are results of the proposed scheme.

In Fig. 3 and Fig. 4, the method without history information shows poor results for MOTHER_DAUGHTER and good results for HALL_MONITOR. The simple method shows acceptable results for MOTHER_DAUGHTER and poor
the previous frame to the current frame with motion information. Our proposed scheme reveals its robustness to the speed of the moving objects due to the proper exploitation of the history information.

6. REFERENCES


5. CONCLUSIONS

In this paper, we propose a new spatio-temporal segmentation algorithm. This algorithm uses history information based on the region tracking method. The region tracking method is the projection of the moving object region in results for HALL_MONITOR. Performances of the two schemes are very dependent on the amount of video object's motion. However, our proposed scheme shows good results for both sequences, and is not dependent on the speed of the video object. In Fig. 4, the method without history information seems to be better than the proposed method at this frame. However, at another frame, the method without history information produces holes in video object. Both methods are comparable for HALL_MONITOR.